

What is claimed is:

- 1 1. A method comprising:
2 identifying a representation of a binary image in a pixel matrix,
3 wherein the pixel matrix comprises a plurality of portions;
4 computing the number of runs for a first portion of the pixel matrix,
5 wherein a run is a maximal sequence of pixels having a predetermined value in
6 the first portion;
7 computing the number of neighboring runs between the first
8 portion and a second portion of the pixel matrix, wherein a neighboring run is a
9 run in which at least one pixel of the run is in the neighborhood of a run in an
10 adjacent portion; and
11 computing the Euler number from the number of runs and the
12 number of neighboring runs.
- 1 2. The method of claim 1, computing the number of runs for a first
2 portion of the pixel matrix further comprising:
3 identifying one or more runs of the first portion;
4 counting the number of runs.
- 1 3. The method of claim 2, computing the number of neighboring runs
2 between the first portion and a second portion of the pixel matrix further
3 comprising:
4 determining a neighborhood size;
5 identifying a second run in the second portion; and
6 determining whether at least one pixel of the second run is in the
7 neighborhood of the run.

1 4. The method of claim 3, computing the Euler number from the
2 number of runs and the number of neighboring runs further comprising:

3 subtracting the number of neighboring runs between the first
4 portion and the second portion from a sum of the number of runs in the first
5 portion and the second portion to arrive at a result; and

6 adding the result to an Euler number for a third portion.

1

2 5. The method of claim 4, computing the Euler number from the
3 number of runs and the number of neighboring runs further comprising:

4 subtracting from the result the number of neighboring runs
5 between the third portion and a prior third portion.

1

2 6. The method of claim 5, identifying the binary image in a pixel
3 matrix further comprising identifying pixels of the predetermined value.

1 7. A system comprising:

2 a run processor to compute a run number, wherein the run number
3 is the number of runs in a portion of a pixel matrix; and

4 a neighboring run processor to compute a neighboring run number,
5 wherein the neighboring run number is the number of neighboring runs between
6 the portion and a second portion of the pixel matrix and the neighboring run
7 processor receives a plurality of signals from the run processor.

1 8. The system of claim 7, further comprising:

2 a second run processor to compute a second run number;

3 a second neighboring run processor to compute a second
4 neighboring run number; and
5 a math processor.

1 9. The system of claim 8, wherein the math processor further
2 comprises:

3 a run adder to add the run number to the second run number;
4 a neighboring run adder to add the neighboring run number to the
5 second neighboring run number; and
6 a subtractor to subtract the run number from the neighboring run
7 number.

1 10. The system of claim 7, wherein the portions comprise rows of the
2 pixel matrix.

1 11. The system of claim 7, wherein the portions comprise columns of
2 the pixel matrix.

12. An article comprising a medium storing software for enabling a
processor-based system to:

1 identify a binary image in a pixel matrix, wherein the pixel matrix
2 comprises a plurality of portions;
3 compute the number of runs for a first portion of the pixel matrix;
4 compute the number of neighboring runs between the first portion
5 and a second portion of the pixel matrix; and

compute an Euler number from the number of runs and the number of neighboring runs.

1 13. The article of claim 12, further storing software for enabling a
2 processor-based system to compute the number of runs for a first portion of the
3 pixel matrix by:

4 identifying one or more runs of the first portion, wherein a run is a
5 maximal sequence of pixels with a predetermined value in the first portion;
 counting the number of runs.

1 14. The article of claim 13, further storing software for enabling a
2 processor-based system to compute the number of neighboring runs between
3 the first portion and a second portion of the pixel matrix by:

4 determining a neighborhood size;
5 identifying a second run in the second portion; and
6 determining whether at least one pixel of the second run is in the
7 neighborhood of the run.

1 15. The article of claim 14, further storing software for enabling a
2 processor-based system to compute the Euler number from the number of runs
3 and the number of neighboring runs by:

4 subtracting the number of neighboring runs between the first
5 portion and the second portion from a sum of the number of runs in the first
6 portion and the second portion to arrive at a result; and
7 adding the result to an Euler number for a third portion.

1 16. The article of claim 15, further storing software for enabling a
2 processor-based system to compute the Euler number from the number of runs
3 and the number of neighboring runs by:
4 subtracting from the result the number of neighboring runs
5 between the third portion and a prior third portion.

11/20/2019 10:00:00 AM